SAMEER MOTWANI
CS 6240-01
Homework 4

Program Discussion

Describe briefly how each step of your program is transforming the data. Be precise, e.g., by showing the structure of the input and output as a table.

Step No.	Description	Input	Output
1	Read the file	Bz2 compressed files	output of this step is an RDD (String,List[String]) indicating pair (pageName, adjacencyList)
2	Filter Invalid lines	RDD	RDD
2	Adding the missing dangling nodes	RDD from Step 1	pair RDD1 with dangling nodes
3	Creating a default page Rank RDD2 and then join it to the (page, adjacencyList) RDD1	RDD2 & RDD1	RDD3
4	Iterate over 10 iterations to calculate page rank for each node	RDD3(Initially) & RDD4(created in step 1) and is joined with RDD3 at the end of iteration	RDD3 with page ranks after each iteration
5	calculate topK nodes	RDD3	Array with page rank and page name using top(100) function

For each step, state if the dependency is narrow (no shuffling) or wide (shuffling). How many stages does your Spark have? (10 points)

Step No.	Description	Dependency
1	Read the file	Narrow
2	Filter Invalid lines	Narrow
2	Adding the missing dangling nodes	wide
3	Creating a default page Rank RDD2 and then join it to the (page, adjacencyList) RDD1	wide
4	Iterate over 10 iterations to calculate page rank for each node	wide
5	calculate topK nodes	wide

To sum up:

1. Transformations with Narrow dependencies in my program:

- map
- mapValues
- flatMap
- filter

2. Transformations with Wide dependencies in my program:

- reduceByKey
- leftOuterJoin

There are 90 stages in total that my spark program has. Please find the screenshot below for reference:

Performance Comparison

Report for both configurations the Spark execution time. For comparison, also include the total execution time (from pre-processing to top-k) of the corresponding Hadoop executions from Assignment 3.

Time comparison of 2 Hadoop EMR runs (Assignment 3):

Number of AWS Machines	Total run time(mins)
6	60.6
11	30

Time comparison of 2 Spark EMR runs (Assignment 4):

Number of AWS Machines	Total run time (mins)
6	50
11	26

Discuss which system is faster and briefly explain what could be the main reason for this performance difference?

As we can see spark is faster, this can be due to:

1. Apache Spark processes data in-memory while Hadoop MapReduce persists back to the disk after a map or reduce action.
2. Spark keeps JVM running in node. hence time is not wasted in spawning of JVM's whereas Map reduce needs to spawn a JVM every time.